The Virtual Machine (VM) Scaler:

Supporting Environmental Modeling for the Cloud Services Innovation Platform (CSIP)

Wes J. Lloyd

July 9, 2014

Colorado State University, Fort Collins, Colorado USA

2014 ESIP Summer Meeting

Outline

- CSIP: Cloud Services Innovation Platform
- Scientific Modeling Cloud Challenges
- The Virtual Machine (VM)-Scaler
  - Introduction
  - Support for Model Services
- Conclusion

CSIP

Cloud Services Innovation Platform

Cloud Services Innovation Platform

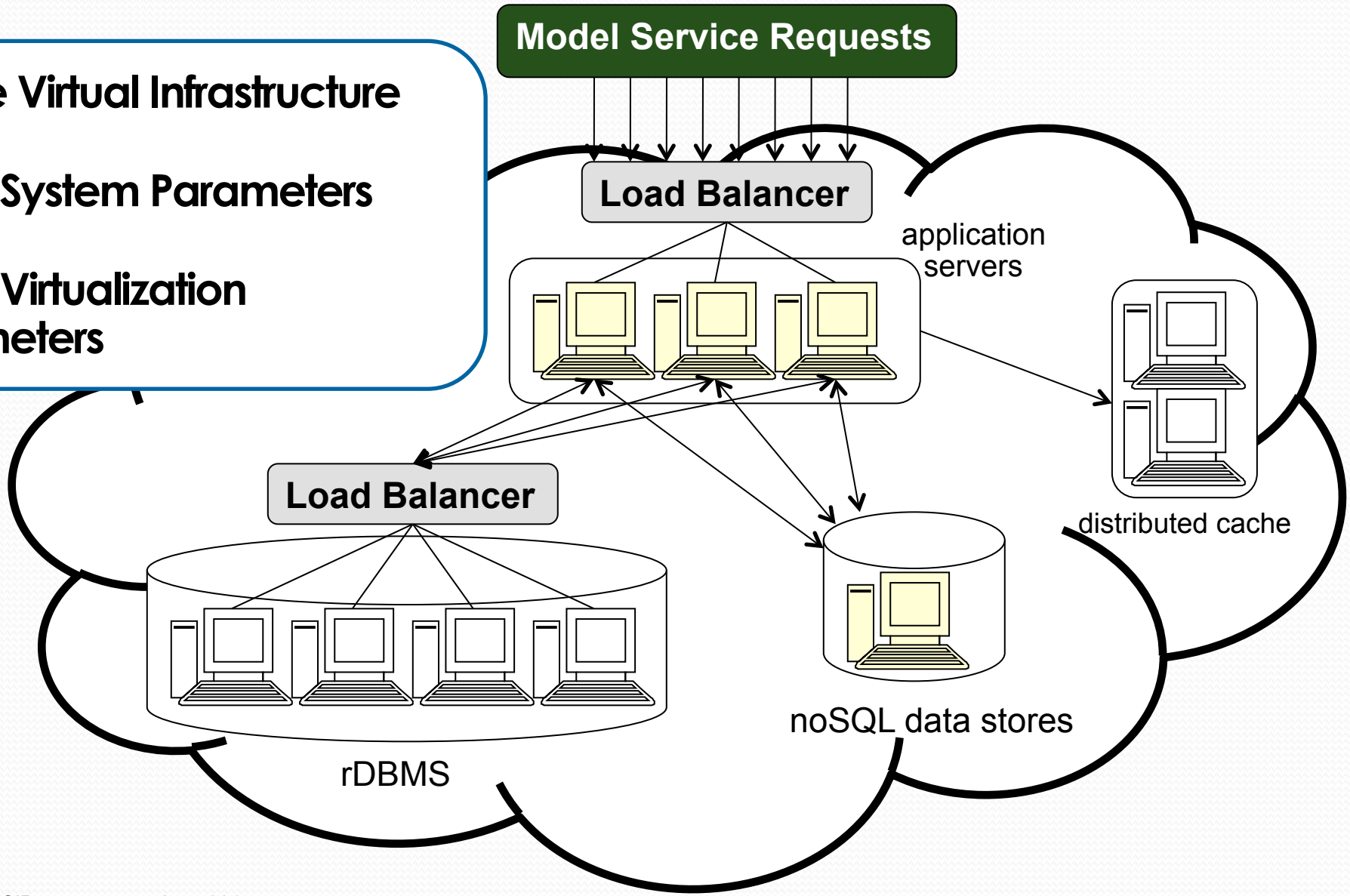
- Provide scientific modeling-as-a-service (MaaS)
- Facilitate science research and delivery
  - Increase availability and throughput of models
  - Scalable cloud infrastructures: private, public, hybrid cloud
  - Cost effective, component based, interoperable
  - Integrated data services supporting model parameterization
  - Auditable with tracking system

Supporting Science Discovery and Delivery

- USDA-ARS: AG Systems Model Research
  - Deploy legacy model services
  - Support new model development
    - Harness scalable cloud resources
    - Support asking new questions: monte carlo, model calibration
- USDA-NRCS: AG Systems Production Models
  - Support Field Offices consultations with farmers
  - Many agency models
  - Deployed as desktop applications
  - Infrequent manual updates

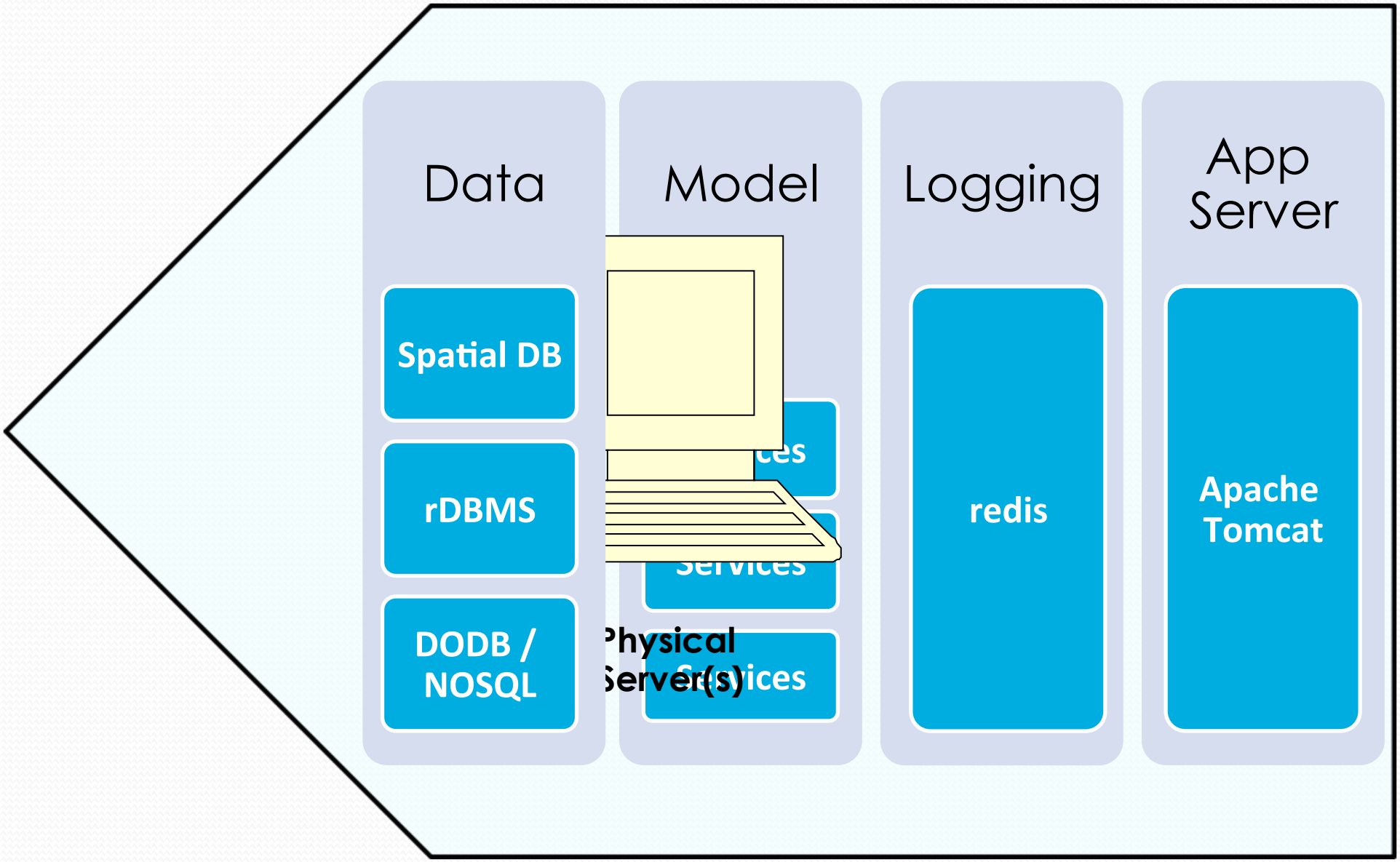
CSIP Model Services

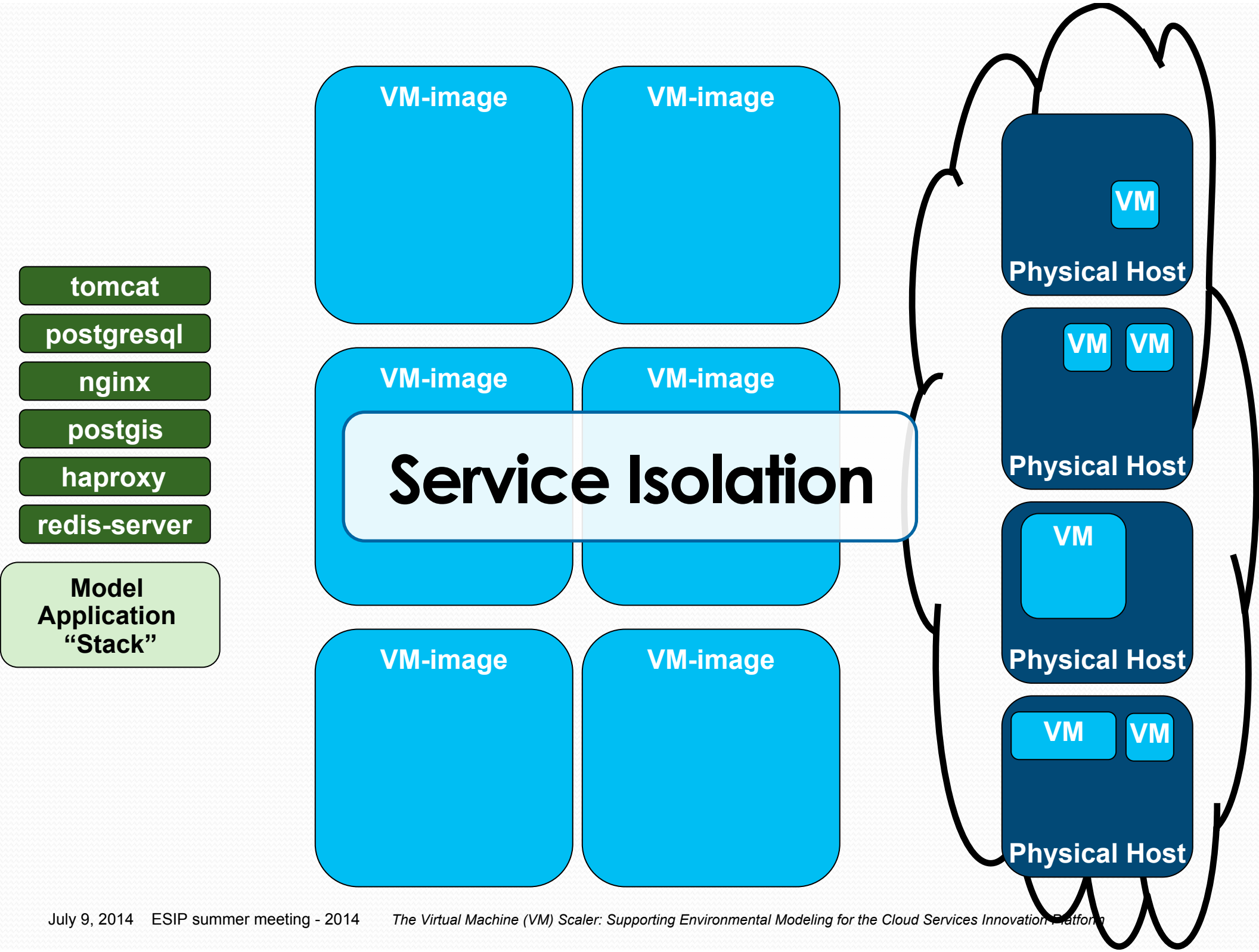
- Scale Virtual Infrastructure
- Tune System Parameters
- Tune Virtualization Parameters

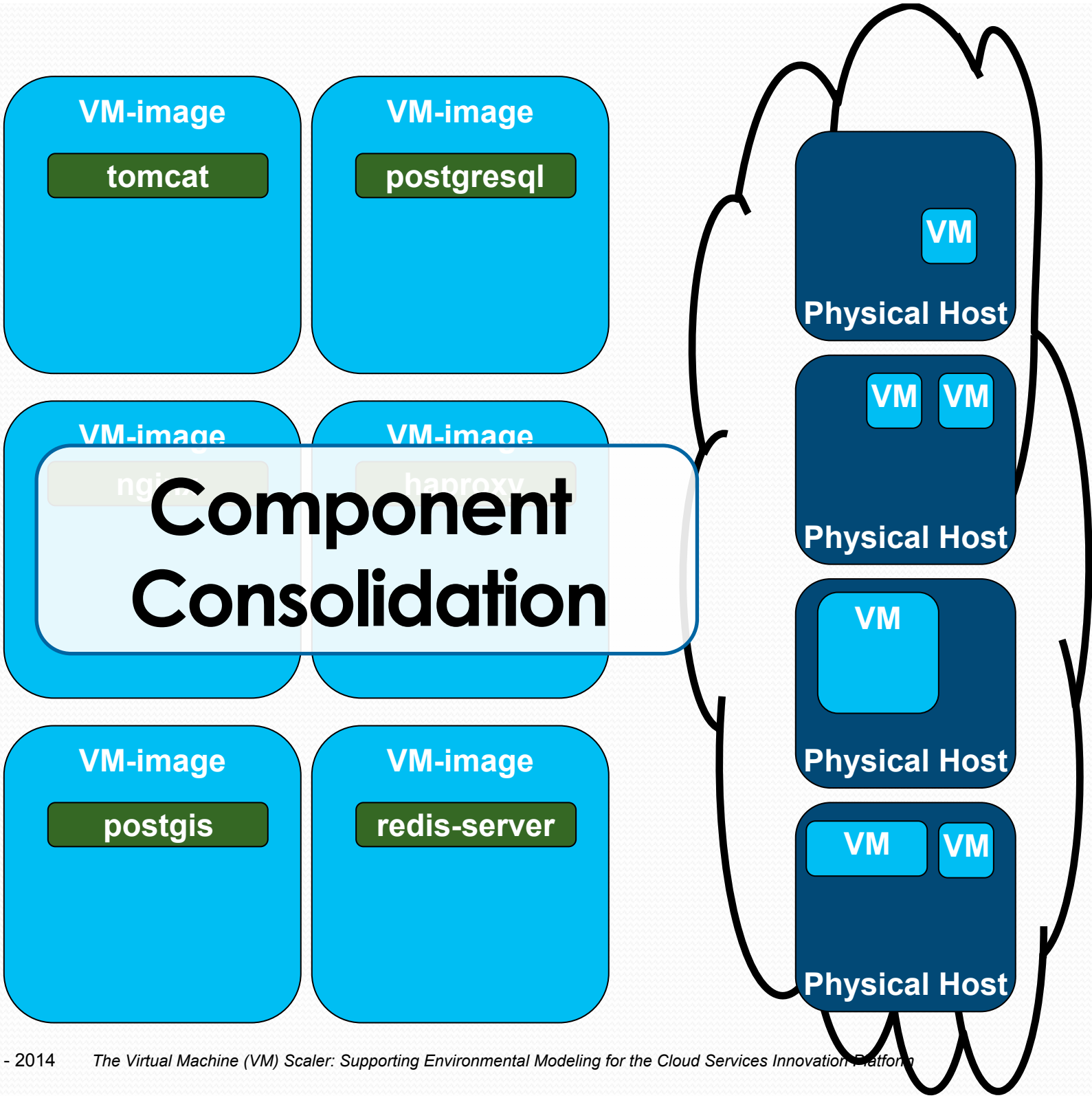


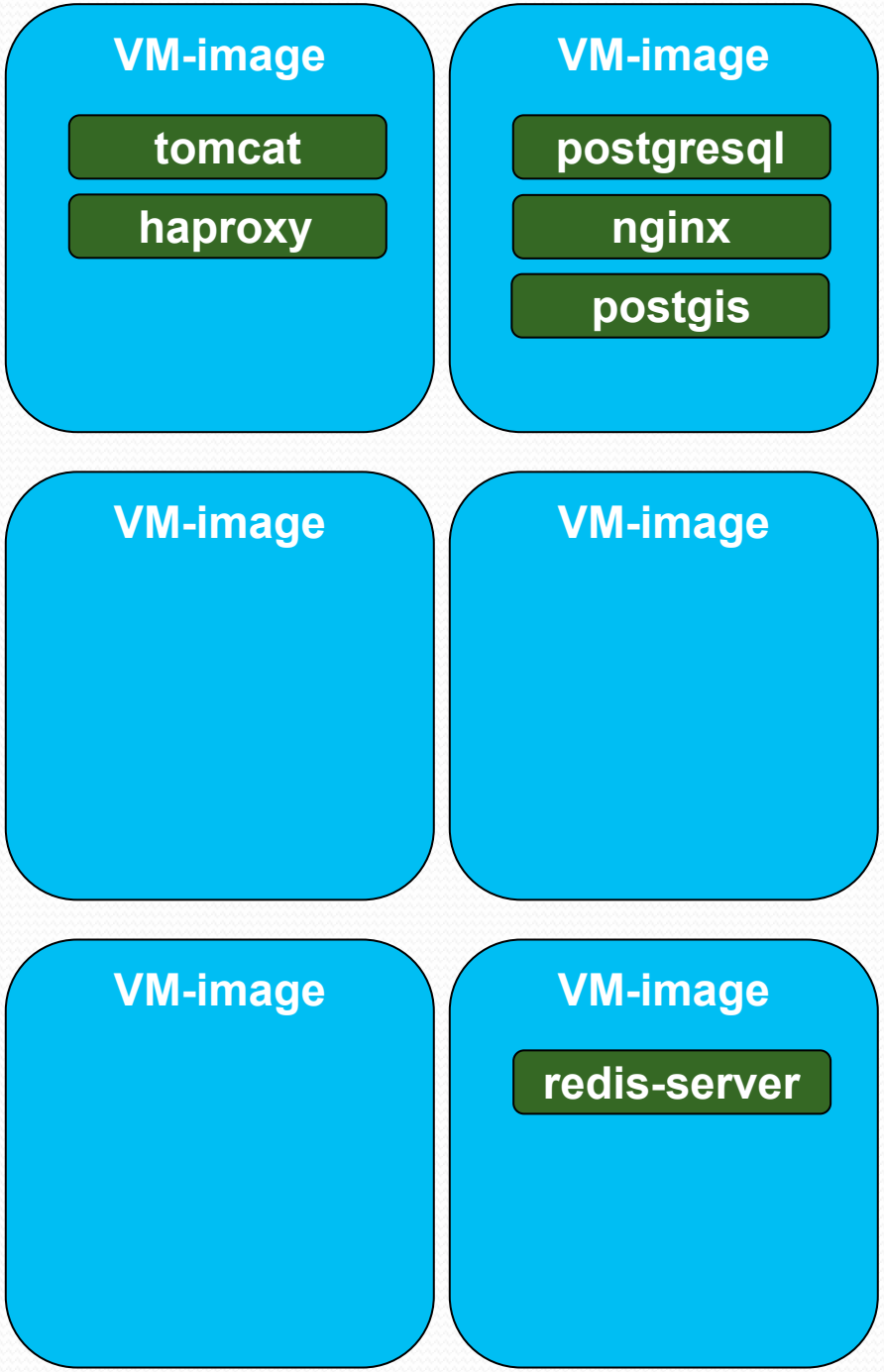
Scientific Modeling Cloud Challenges

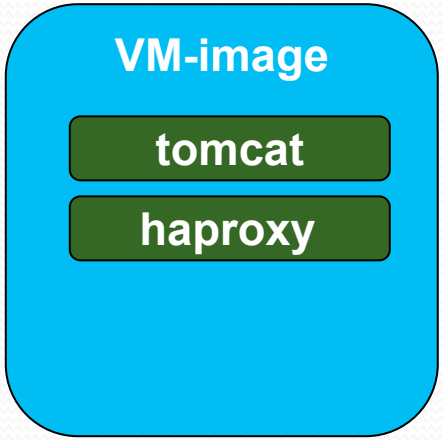
Model Services Deployment

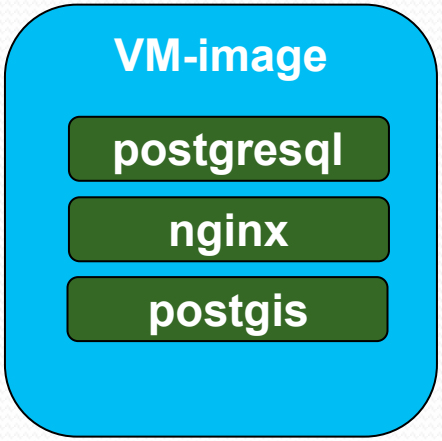


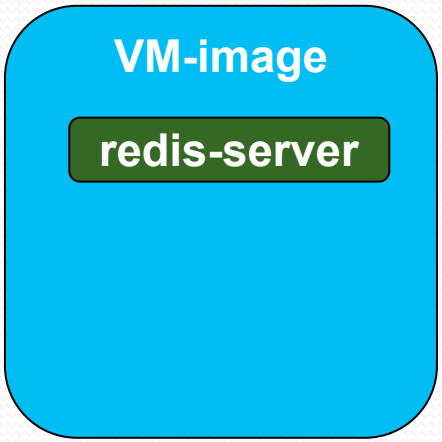

tomcat
postgresql
nginx
postgis
haproxy
redis-server
Model Application "Stack"
VM-image
VM-image
VM-image
VM-image
VM-image
VM-image
Service Isolation
VM
Physical Host
VM
VM
Physical Host
VM
Physical Host
VM
VM
Physical Host

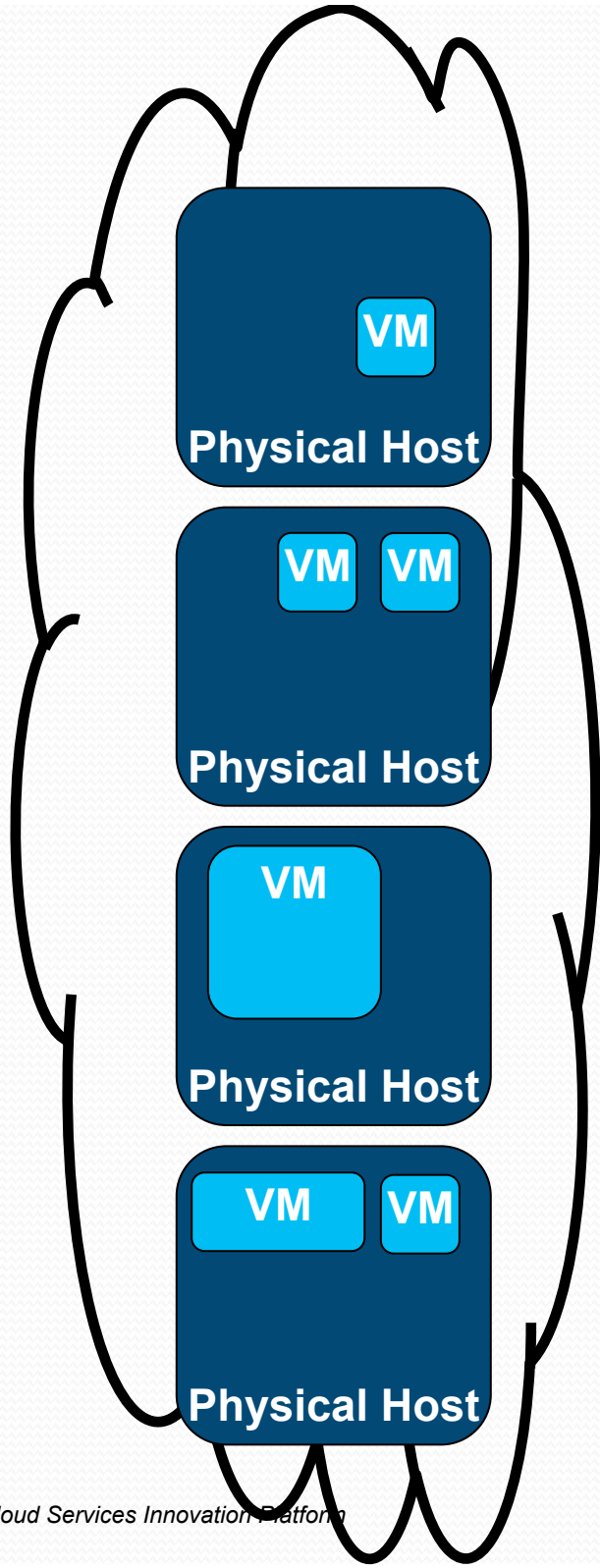

VM-image
tomcat
VM-image
postgresql
VM-image
VM-image
Component
Consolidation
VM-image
postgis
VM-image
redis-server
VM
Physical Host
VM
VM
Physical Host
VM
Physical Host
VM
VM
Physical Host

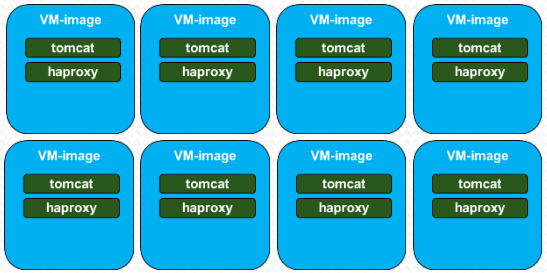

VM-image
tomcat
haproxy
VM-image
postgresql
nginx
postgis
VM-image
VM-image
VM-image
redis-server
VM
Physical Host
VM
VM
Physical Host
VM
Physical Host
VM
VM
Physical Host


VM-image
tomcat
haproxy


VM-image
postgresql
nginx
postgis

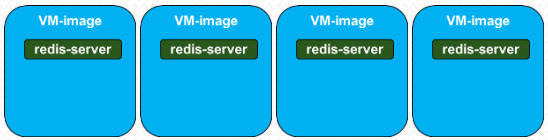

VM-image
redis-server

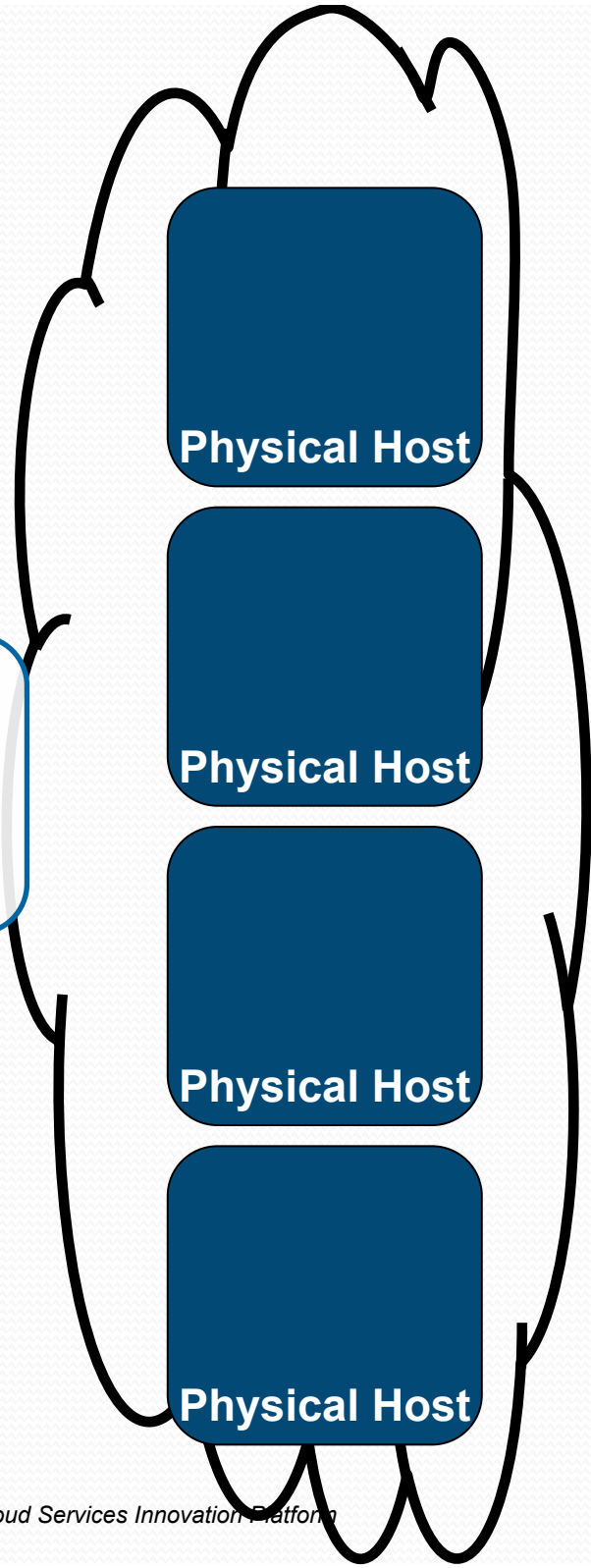

VM
Physical Host
VM
VM
Physical Host
VM
Physical Host
VM
VM
Physical Host



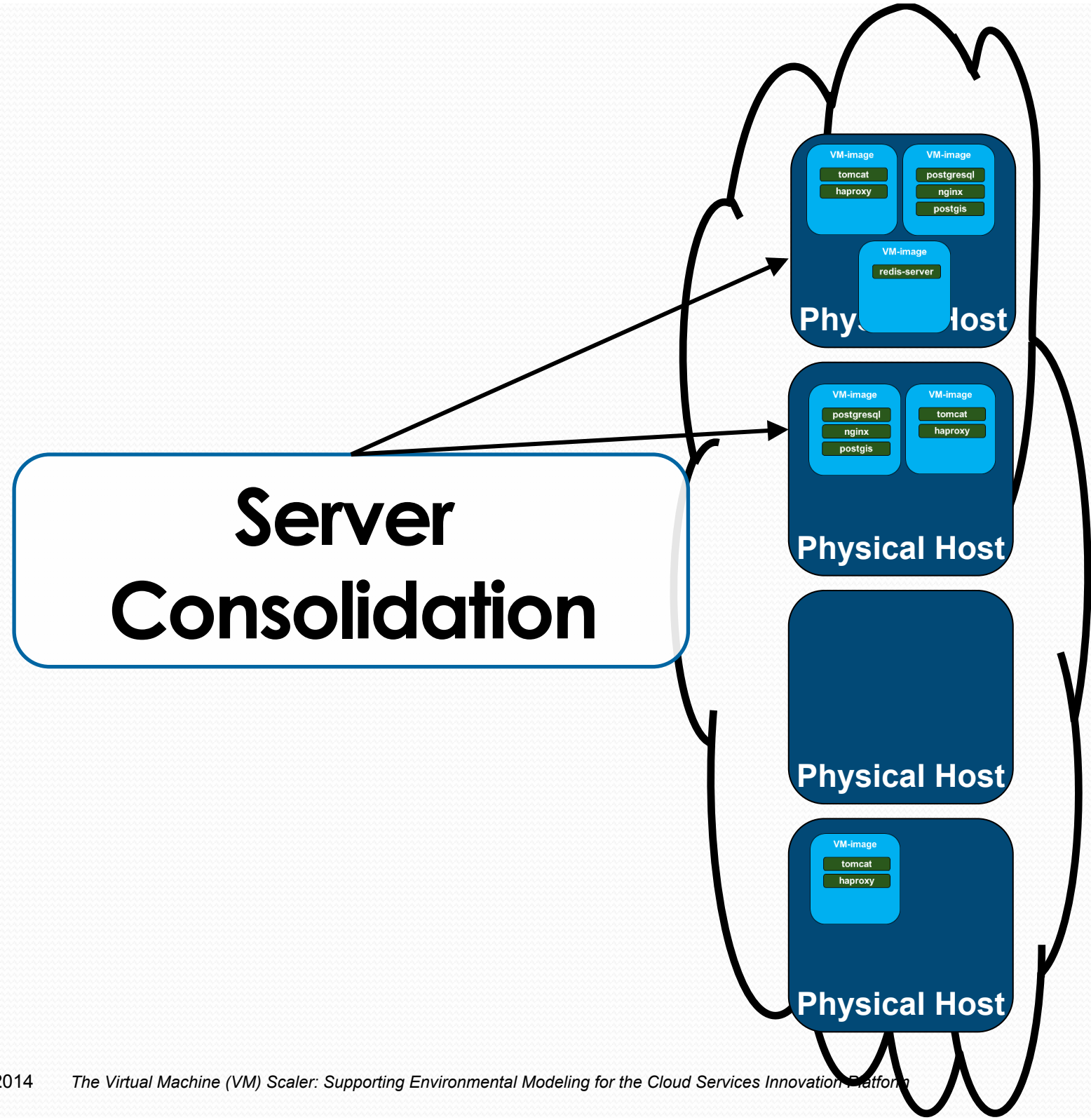
Elasticity

Provisioning Variation

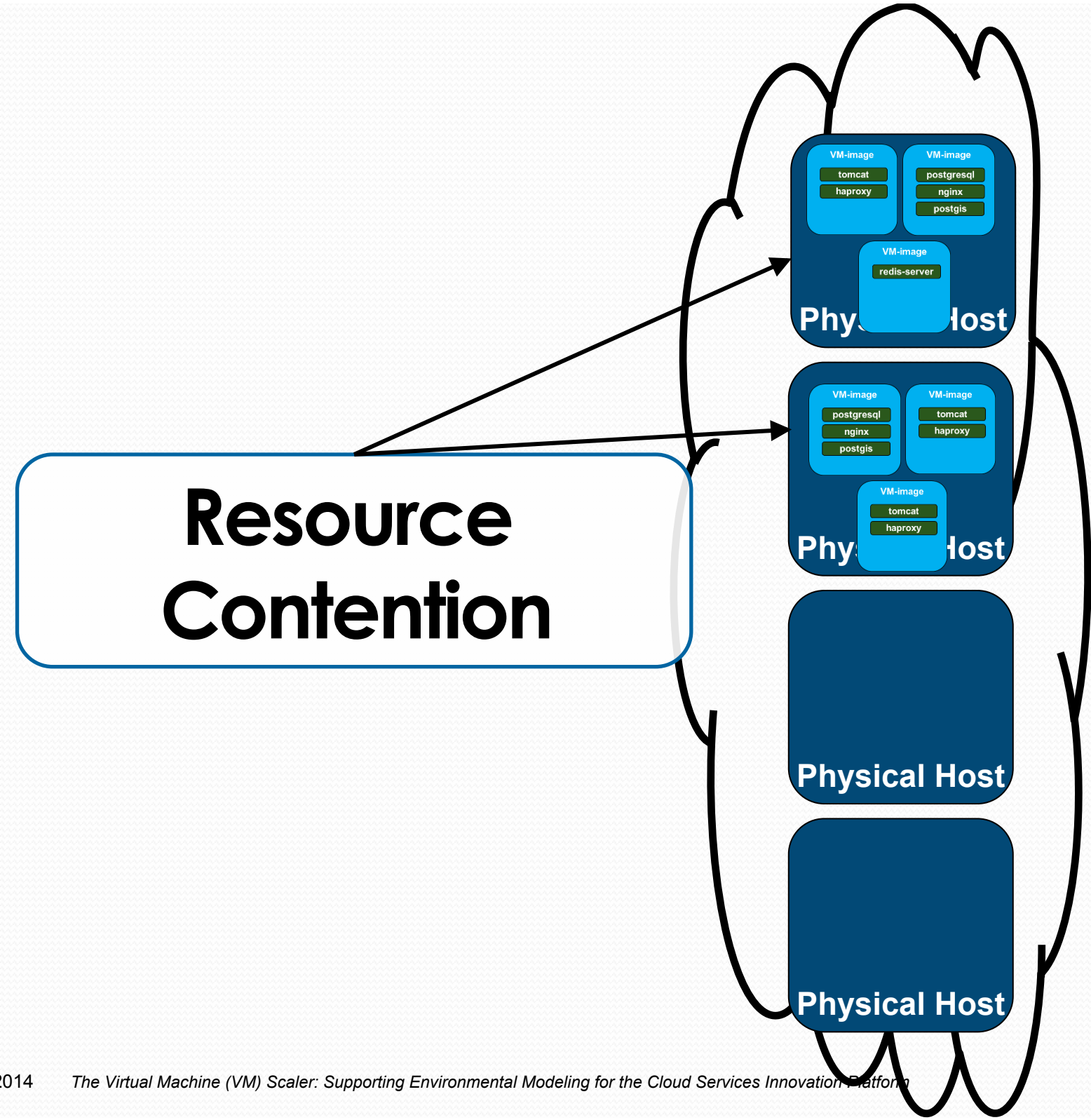




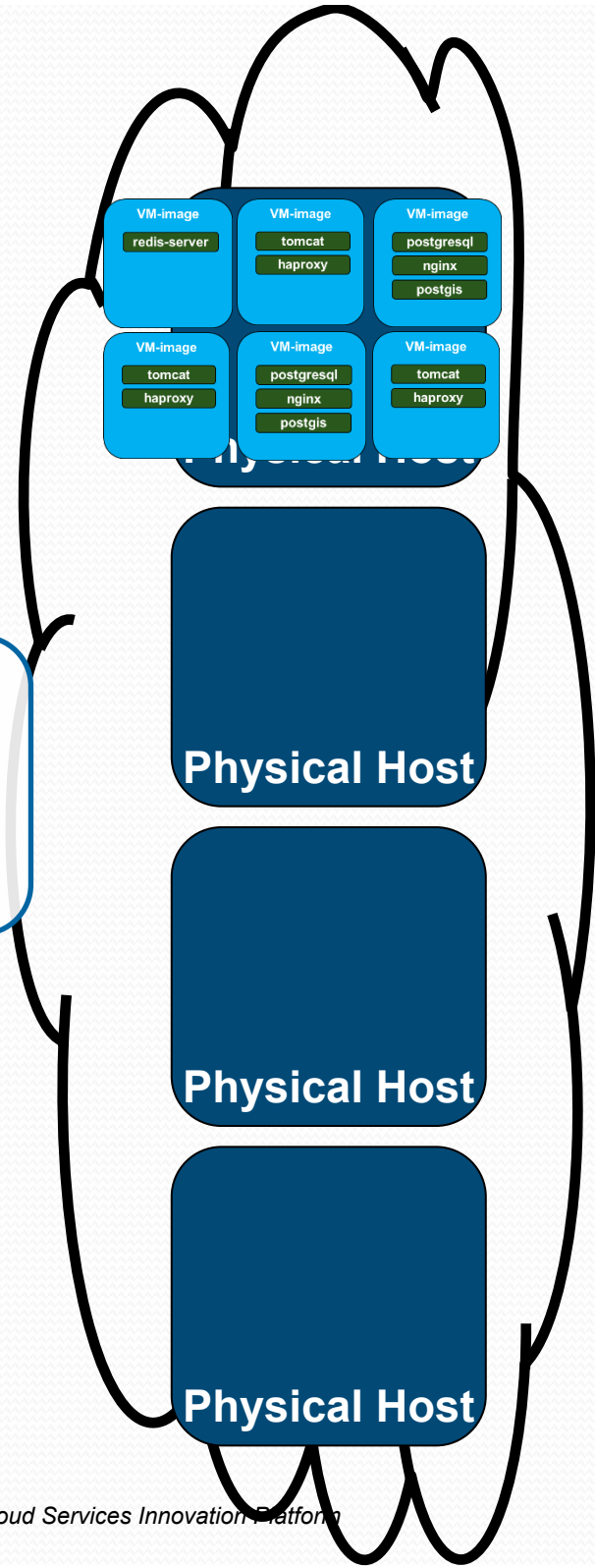
Server Consolidation



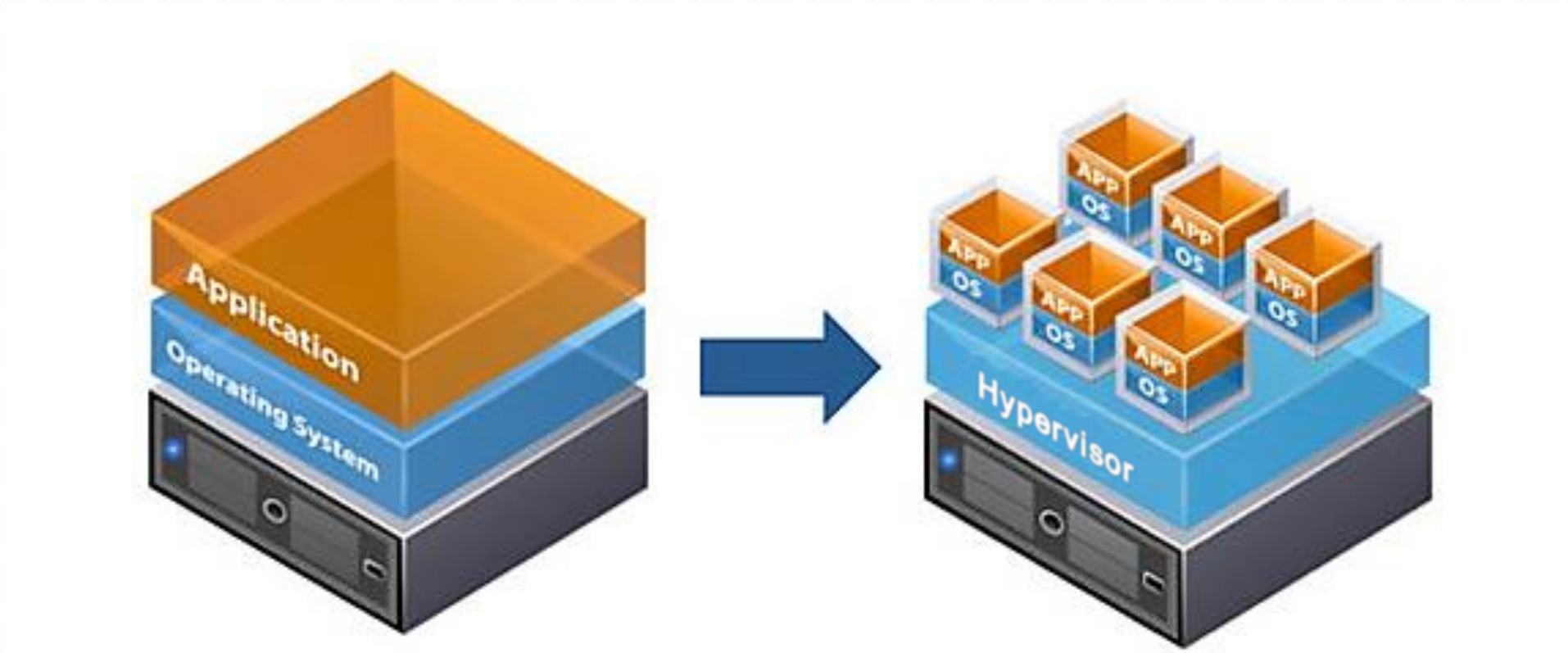
Resource Contention



Overprovisioning



Virtualization Overhead

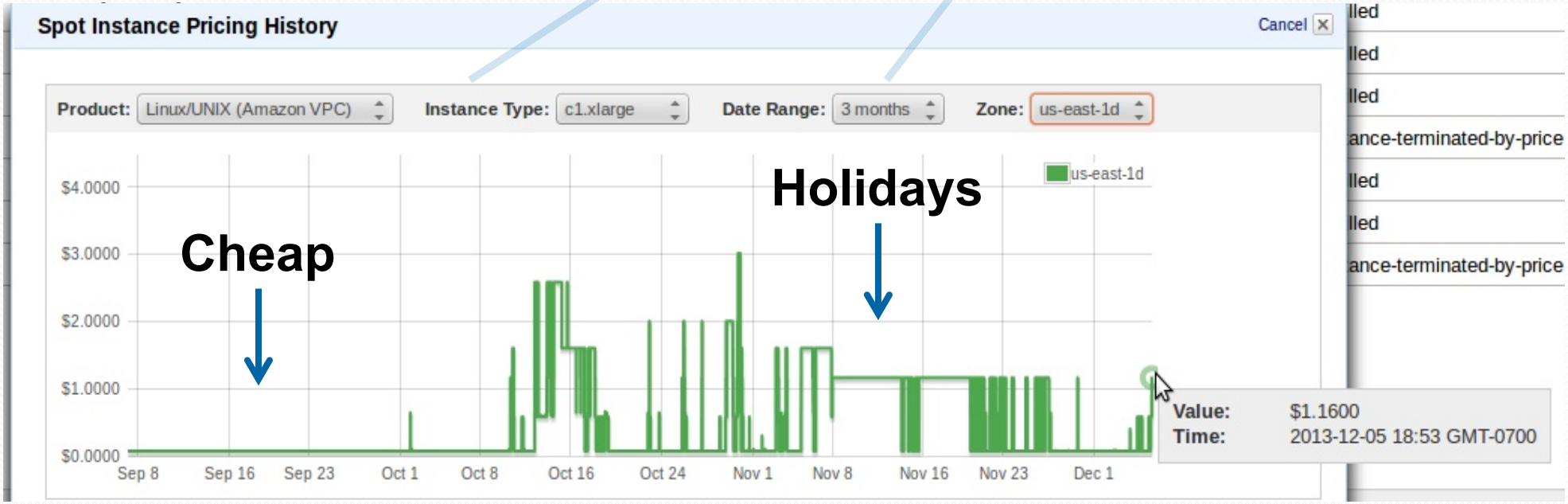


Traditional Architecture **Virtual Infrastructure**

Amazon Spot Instances

c1.xlarge dedicated instances 48¢/ hour (recently reduced!)
Look what happened to pricing…

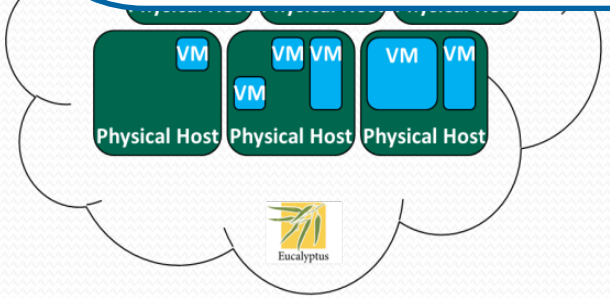
- Cloud Virtual Machines
- Auction based pricing mechanism
- Leverage unused cloud capacity at low cost
- Must manage modeling costs in response to market

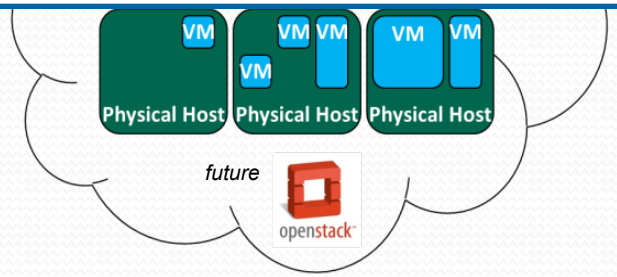


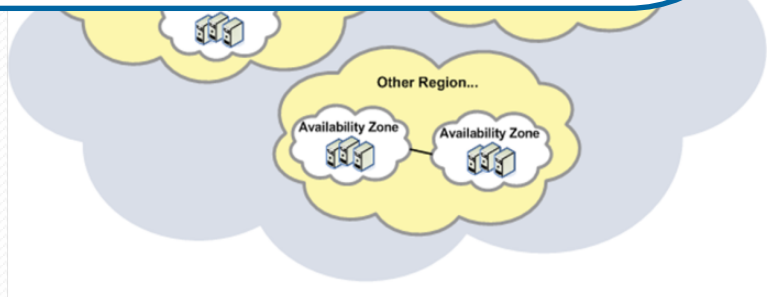
The Virtual Machine (VM) Scaler:

VM-Scaler

- Web services application
  - Rest-based/JSON
  - Harnesses EC2/Eucalyptus API
  - Manages virtual cloud infrastructure
  - Supports scientific modeling-as-a-service
  - Supports Amazon, Eucalyptus 3.x clouds
  - Extensible to others, e.g. OpenStack







VM-Scaler: Supporting Features

- VM Pools
- Resource utilization data collection
- Resource utilization checkpointing
- Scaling Tasks
- Hot spot detection
- LeastBusy VM Placement
- LeastBusy Job Scheduling

VM Pools

- Supports work with many same-type VMs
  - Size enforcement: minimum / maximum
  - State: available / in-use
  - Operations: execute common scripts across the pool
- Addresses launch latency
  - Public cloud
  - Spot instances
- Supports VM recycling

Resource Utilization Data Collection

- Resource utilization sensors
  - Sensor on each VM/PM
  - Transmits data to VM-Scaler @ configurable intervals

Disk
- dsr: disk sector reads
- dsreads: disk sector reads completed
- drm: merged adjacent disk reads
- readtime: time spent reading from disk
- dsw: disk sector writes
- dswrites: disk sector writes completed
- dwm: merged adjacent disk writes
- writetime: time spent writing to disk

CPU
- CPU time
- cpu usr: CPU time in user mode
- cpu krn:CPU time in kernel mode
- cpu_idle: CPU idle time
- contextsw: # of context switches
- cpu_io_wait: CPU time waiting for I/O
- cpu_sint_time: CPU time serving soft interrupts
- loadavg: (# proc / 60 secs)

Network
- nbr: network bytes sent
- nbs: network bytes received

Resource Utilization Checkpointing

- Captures resource utilization at time (T_0, T_1, T_2, T_3)
  - Assigned a name
  - Used to calculate resource utilization Δ from time T_x
  - Synchronized samples used (within 1 second)
- Supports capturing total resource utilization for a model workloads executed across a pool of VMs
  - Characterizes: *CPU, disk I/O, network I/O* utilization
  - Supports dynamically scaled VM pools

Scaling Tasks

- Scaling service request
  - Request to scale a specific application tier
- Prelaunch VMs: preprovision fixed # VMs (pools)
- Sequential VM launch (increase by 1)
- Parallel many-VM launch (increase by many)
  - *Future: Dynamic # of new VMs based on hot spot
- Handle launch failure
  - Reschedule VM launches for timeouts and DOA VMs

Hot Spot Detection

- Resource utilization thresholds
  - Scale when thresholds exceeded
    - MAX CPU time, MIN CPU idle time, # of CPU context switches
  - Initial VM: evenly distributed homogeneous workloads
  - Average of VMs: heterogeneous workloads

- Performance model approach
  - Predict future resource deficits
  - Requires scaling training dataset
  - Tests completed with multiple linear regression

Exponential moving average

$$\mathbf{F_{t+1} = \alpha A_t + (1-\alpha) F_t}$$

Least-Busy VM Placement

- RU-sensors collect Virtual/ Physical machine data @ fixed intervals
- Busy-Metric used to calculate aggregate load at each physical machine
  - Flexible metric design
  - Objective not to design perfect metric / VM scheduler

Resource Utilization Data

CPU
- Total CPU time
  weighted 2X

Disk
- Disk sector reads (DSR)
- Disk sector writes (DSW)

Network
- Network bytes sent (NBR)
- Network bytes received (NBS)

Virtualization
- Total VM count per host

Least-Busy Job Placement

- LeastBusy
  - Distribute modeling requests to VM (PM) with lowest resource utilization
  - Harnesses resource utilization data
  - Application agnostic
  - Alternative to traditional load balancer
    - round-robin, least-connection job placement, eg. haproxy

Conclusion

- For more information:

Wes Lloyd

Email: wlloyd@acm.org

Colorado State University

http://www.cs.colostate.edu/~wlloyd/

Questions